Attorney Docket: 044204-0308162
Client Reference:

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: JASON KIM, ET AL.

Application No.: 09/847,991 Group No.: 2112

Filed: May 2, 2001 Examiner: MYERS, Paul R.

Title: CROSS BAR MULTIPATH RESOURCE CONTROLLER SYSTEM AND METHOD

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

ATTENTION: Board of Patent Appeals and Interferences

APPELLANTS' BRIEF (37 C.F.R. §41.37)

Sir:

This paper is further to the Notice of Appeal dated February 9, 2006, for which a supportive brief was due April 9, 2006.

Petition for Extension of Time:

Applicant hereby petitions for a 3-month extension of time, extending the period for filing the Brief from April 9, 2006 until present. The Commissioner is authorized to charge Deposit Account 033975 (order no. 044204-0308162) for the requisite 3-month small-entity extension fee of **$510.00**.

Other Fees:

The Commissioner is authorized to charge the small entity fee for filing a brief in support of an appeal in the amount of $250.00, and any required fee to Pillsbury Winthrop Shaw Pittman LLP's deposit account no. 03-3975 (order no. 044204-0308162).

I. REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: PortalPlayer, Inc.

II. RELATED APPEALS AND INTERFERENCES

No other appeals or interferences will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: **27**

B. STATUS OF ALL THE CLAIMS IN APPLICATION

Claims 1-27 are pending in the Application and have been rejected.

Claims 1-27 have been rejected.

No claims have been cancelled.

C. CLAIMS ON APPEAL

The rejections of claims 1-27 are being appealed.

IV. STATUS OF AMENDMENTS

No amendments to the claims were submitted or made in the Application after the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The claims of the present Application are directed to cross bar multipath resource controller systems and methods that permit multiple processors in a computer system to independently access different resources in the computer system simultaneously. Abstract. In the Application, claims 1, 11, 21, 24 and 27 are independent claims. Figure 2 of the Application is presented below to assist description.

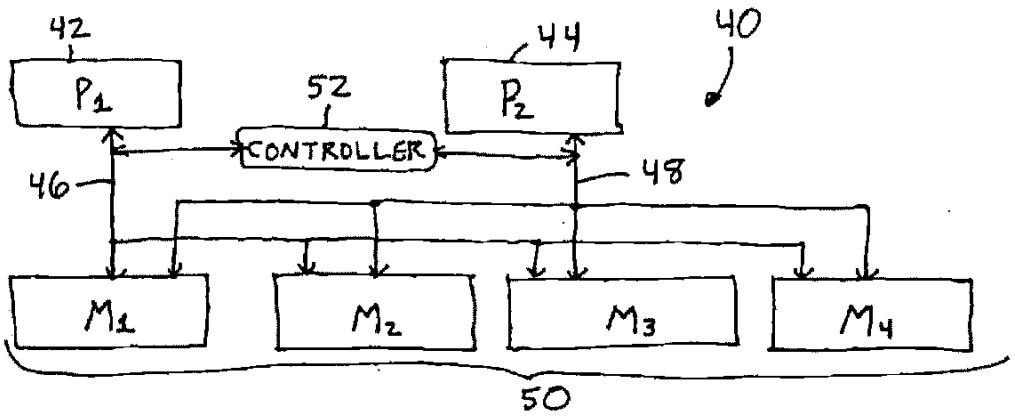


FIGURE 2

Claim 1 is directed to a computer system having a multipath cross bar bus and comprises one or more processors 42 and 44, one or more resources 50 capable of being shared by the one or more processors 42 and 44, and a resource controller 52 and bus that is connected to each resource 50 and to each processor 42, 44, wherein the resource controller 52 is capable of permitting each processor 42, 44 to simultaneously access a different resource from the one or more resources 50, and wherein the resource controller 52 includes a hardware semaphore unit (not shown) for controlling access to the shared resources 50. *See* Specification, page 6, line 6 to page 7, line 7.

Claim 11 is directed to an apparatus for controlling access to one or more computing resources 50 by one or more processors 42 and 44 comprising a resource controller 52 and bus that is connected to each resource 50 and to each processor 42, 44 wherein the resource controller 52 is capable of permitting each processor 42, 44 to simultaneously access a different resource from the one or more resources 50, and wherein the resource controller 52 includes a hardware semaphore unit (not shown)for controlling access to the one or more resources 50. *See* Specification, page 6, line 6 to page 7, line 7.

Claim 21 is directed to an apparatus for controlling access to one or more memory resources 50 by one or more processors 42 and 44 comprising a memory resource controller 52 and bus that is connected to each memory resource 50 and to each processor 42, 44, wherein the resource controller 52 includes a hardware semaphore unit (not shown) for controlling access to the shared resources 50, and wherein the memory resource controller 52 is capable of permitting each processor 42, 44 to simultaneously access a different resource from the one or more memory resources 50. *See* Specification, page 6, line 6 to page 7, line 7.

Claim 24 is directed to an apparatus for controlling access to one or more peripheral resources 50 comprising a peripheral resource controller 52 and bus that is connected to each peripheral resource 50 and to each processor 42, 44, wherein the resource controller 52 is capable of permitting each processor 42, 44 to simultaneously access a different peripheral resource from the one or more peripheral resources 50, wherein the peripheral resource controller 52 includes a hardware semaphore unit (not shown) for controlling access to the one or more peripheral resources 50, the hardware semaphore unit being configured to receive requests from the one or more processors 42 and 44 and prioritize access based on the requests. *See* Specification, page 6, line 6 to page 7, line 7.

Claim 27 is directed to a computer system comprising a first processor 42 capable of executing a set of instructions, a second processor 44 capable of executing a set of instructions, a multipath memory controller 52 having a first bus 46 that is capable of connecting the first processor 42 to a set of memory resources 50 and a second bus 48 that is capable of connecting the second processor 44 to the same set of memory resources 50 wherein the first and second processors 42 and 44 are capable of simultaneously accessing different memory resources, wherein the multipath memory controller includes a first semaphore unit (not shown) for prioritizing access to the set of memory resources, and a multipath peripheral controller having a first bus that is capable of connecting the first processor 42 to a set of peripheral resources and a second bus that is capable of connecting the second processor 44 to the same set of peripheral resources wherein the first and second processors are capable of simultaneously accessing different peripheral resources, wherein the multipath peripheral controller includes a second semaphore unit for prioritizing access to the set of peripheral resources, wherein at least one of the semaphore units comprises a hardware semaphore unit (not shown). *See* Specification, page 6, line 6 to page 7, line 7.

VII. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The claims in the Application have been rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,636,933 to MacLellan et al. (MacLellan) in view of U.S. Patent No. 5,394,551 to Holt et al. (Holt); over U.S. Patent No. 5,053,942 to Srini (Srini) in view of Holt; over U.S. Patent No. 6,125,429 to Goodwin et al. (Goodwin) in view of Holt; and over U.S. Patent No. 5,081,575 to Hiller et al. (Hiller) in view of Holt and further in view of Goodwin. The Examiner also relies on U.S. Patent No. 5,805,030 to Dhuey et al in rejecting certain of the claims.

At issue in this appeal are the following:

- whether MacLellan teaches or suggests a resource controller capable of permitting one or more processors to simultaneously access different resources;
- whether MacLellan and Holt render obvious a resource controller that includes a hardware semaphore unit for controlling access to shared resources;
- whether Srini and Holt render obvious a resource controller that includes a hardware semaphore unit for controlling access to shared resources;
- whether Hiller and Holt render obvious a resource controller that includes a hardware semaphore unit for controlling access to shared resources;
- whether Goodwin and Holt render obvious a resource controller that includes a hardware semaphore unit for controlling access to shared resources; and
- whether any combination of the cited art render obvious a hardware semaphore unit for controlling access to shared resources.

VIII. ARGUMENT

The rejections of the claims should be reversed because the cited art does not teach or suggest each and every limitation of the claims. A *prima facie* case of obviousness under § 103 requires that each and every limitation be taught or suggested in the cited prior art. MPEP 2143.03; *In re Royka*, 490 F.2d 981 (CCPA 1974). The cited prior art does not teach or suggest a resource controller and bus that is connected to each of one or more resources and to each of one or more processors, wherein the resource controller is capable of permitting each processor to simultaneously access a different resource from the one or more resources. In addition the cited prior art does not teach a resource controller that includes a hardware semaphore unit for controlling access to the shared resources.

This appeal is made necessary due to repeated rejections made for repeatedly flawed reasons based on misapprehension of the prior art cited in the various Office Actions. This application has been pending **for more than five years** since its filing date of May 2, 2001.

MacLellan Does Not Teach Or Suggest A Resource Controller Capable Of Permitting One Or More Processors To Simultaneously Access A Different Resource

The claims require a resource controller that is connected to each of one or more resources and to each of one or more processors, wherein the resource controller is capable of permitting each processor to simultaneously access a different resource from the one or more resources. MacLellan does not teach or suggest such resource controller. Nevertheless, the Examiner alleges that message network 260 in FIG. 2 of MacLellan (reproduced below) anticipates the resource controller merely because message network 260 includes a controller and crossbar switch. However, a careful review of MacLellan reveals that message network 260 is explicitly taught as managing message traffic between front-end directors and backend directors only. Nothing in MacLellan teaches or suggests that MacLellan's message network 260 can control, permit or prevent simultaneous access between processors and resources. In contrast, and as will be shown below, MacLellan teaches that messaging network 260 is incapable of controlling, permitting or preventing simultaneous access between processors and resources. Consequently, Appellants respectfully submit that the Examiner erred in rejecting the claims.

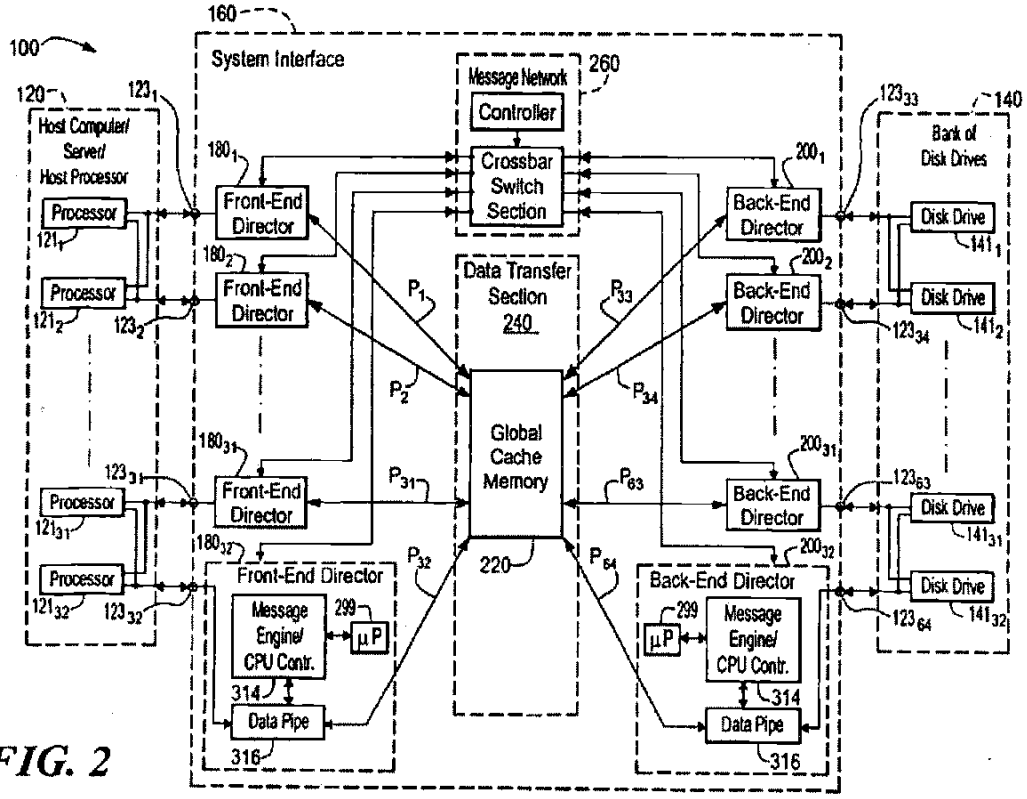


FIG. 2

MacLellan expressly teaches "a messaging network 260 operative independently of the data transfer section 240." MacLellan, col. 5, lines 2-8 *and see* col. 5, lines 35-40, col. 6, lines 34-43 and FIG. 2. Moreover, MacLellan explicitly teaches that "front-end and back-end ***directors 180$_1$-180$_{32}$, 200$_1$-200$_{32}$ control data transfer*** between the host computer/server 120 and the bank of disk drives 140 in response to messages passing between the directors 180$_1$-180$_{32}$, 200$_1$-200$_{32}$ through the messaging network 260." MacLellan, col. 5, lines 14-19 (emphasis added). According to MacLellan then, directors 180$_1$-180$_{32}$, 200$_1$-200$_{32}$ control data transfer between processors and disk drives and messaging network 260 is used only to communicate messages between these instrumentalities of data transfer. This teaching directly contradicts the Examiner's equation of messaging network 260 with the requisite resource controller capable of permitting each processor to simultaneously access a different resource from the one or more resources. *See* Office Action of 11/9/05, page 5, last paragraph.

In other passages throughout MacLellan, the Examiner's error is confirmed. For example, MacLellan states:

> The mapping of which back-end directors 200_1-200_{32} control which disk drives 141_1-141_{32} in the bank of disk drives 140 is determined during a power-up initialization phase. The map is stored in the global cache memory 220. Thus, when the front-end director 180_1-180_{32} makes a request for data from the global cache memory 220 and determines that the requested data is not in the global cache memory 220 (i.e., a "miss"), the front-end director 180_1-180_{32} is also advised by the map in the global cache memory 220 of the back-end director 200_1-200_{32} responsible for the requested data in the bank of disk drives 140. The requesting front-end director 180_1-180_{32} then must make a request for the data in the bank of disk drives 140 from the map designated back-end director 200_1-200_{32}.

MacLellan, col. 6, lines 20-29. Thus, MacLellan explicitly teaches that global cache memory 220 identifies back-end directors to front-end directors to enable front-end directors to access data not available in cache. MacLellan also teaches that:

> ...the front-end director 180_1-180_{32} sends a message to the appropriate one of the back-end directors 200_1-200_{32} through the message network 260 to instruct such back-end director 200_1-200_{32} to transfer the requested data from the bank of disk drives 140 to the global cache memory 220. When accomplished, the back-end director 200_1-200_{32} advises the requesting front-end director 180_1-180_{32} that the transfer is accomplished by a message, which passes from the back-end director 200_1-200_{32} to the front-end director 180_1-180_{32} through the message network 260. In response to the acknowledgement signal, the front-end director 180_1-180_{32} is thereby advised that such front-end director 180_1-180_{32} can transfer the data from the global cache memory 220 to the requesting host computer processor 121_1-121_{32} as described above when there is a cache "read hit".

MacLellan, col. 6, lines 49-65. It is readily apparent that, in MacLellan, processors and disk drives are connected indirectly to cache 220 through respective directors, that the directors control data transfer, and that message network 260 is merely a conduit for communication between directors. MacLellan teaches that processors are permitted access to cache by front-end directors 180_1-180_{32}, while disk drives are permitted access to cache by back-end directors 200_1-200_{32}. A processor receives data from front-end directors 180_1-180_{32} after the data is placed in cache 220 by back-end directors 200_1-200_{32}. No connection is provided between any of processors, disk drives and message network 260. No direct access between MacLellan's messaging network 260 and global cache 220 is contemplated and data transfer section 220 and messaging network 260 are explicitly maintained independent from one another. Messaging

network 260 merely facilitates communications between front-end and back-end directors 180_1-180_{32}, 200_1-200_{32} and does not facilitate any communications between processor and disk drives. Therefore, MacLellan does not teach or suggest that message network 260 is connected to each of one or more resources and to each of one or more processors, or that message network 260 is capable of permitting each processor to simultaneously access a different resource from the one or more resources.

Moreover, the above cited passages show that MacLellan describes an asynchronous system in which simultaneous access of disk drives by processors is neither possible nor necessary. In MacLellan, processors access to data in cache is provided independently of disk drive access to the same data in cache and no data flows directly between processor and disk drive. Therefore, it cannot be said that MacLellan teaches simultaneous access of processors to different resources because no direct access is provided and access is expressly asynchronous. Nor can any of the cited art cure these deficiencies.

Accordingly, the Examiner erred in asserting that the messaging network 260 of MacLellan teaches or suggests that is connected to each of one or more resources and to each of one or more processors, wherein the resource controller is capable of permitting each processor to simultaneously access a different resource from the one or more resources.

Therefore, the rejections of the claims should be reversed

MacLellan And Holt Do Not Render Obvious A Resource Controller That Includes A Hardware Semaphore Unit For Controlling Access To Shared Resources

MacLellan does not teach or suggest a resource controller that includes a hardware semaphore unit for controlling access to the shared resources as required in the claims. Nevertheless, the Examiner alleges that a superficial reference to data communication semaphores in MacLellan provides sufficient motivation for a skilled artisan to combine MacLellan with Holt in a manner that renders obvious a hardware semaphore unit for controlling access to shared resources. The Examiner is wrong.

As demonstrated *supra*, MacLellan teaches an asynchronous system in which a message is sent through a messaging network 260 to a back-end director 200_1-200_{32} identified by a map in a global cache 220; the back-end director 200_1-200_{32} populates global cache 220 with data and sends a message when complete. *See* MacLellan, col. 6, lines 20-29 and 49-65. Thus, the messaging network 260 is for controlling message traffic, not for controlling access to the shared resources

(disk drives) by processors. Processors in MacLellan do not have direct access to the shared resources since a global cache is interposed between directors 180_1-180_{32} and 200_1-200_{32} which are interposed between processors and disk drives, respectively. Messaging network 260 merely facilitates communications between front-end and back-end directors 180_1-180_{32}, 200_1-200_{32} and messaging network 260 does not facilitate communications between processor and disk drives.

Semaphores are mentioned in MacLellan in context of a description of the content of a serial data communications bitstream that describes the content of serial information stream as including "protocol signaling (e.g. semaphore)." MacLellan, col. 14, lines 25-45. MacLellan never teaches or suggests semaphores for controlling access to shared resources. MacLellan never teaches or suggests a hardware semaphore unit. MacLellan teaches a system in which processors do not access disk drives directly and simultaneous access is not taught or suggested. Consequently a skilled artisan would not have been motivated to combine and modify the teachings of MacLellan and Holt such that a hardware semaphore unit for controlling access to the shared resources would have been rendered obvious.

Therefore, the Examiner erred in rejecting the claims based on MacLellan and Holt.

Srini And Holt Do Not Render Obvious A Resource Controller That Includes A Hardware Semaphore Unit For Controlling Access To Shared Resources

Srini is silent regarding use of semaphores to control access to shared resources. A careful reading of Srini yields a description of a system that uses a fairness-based arbitration scheme based on a state machine. One skilled in the art would have easily recognized that the operation of the state machine is incompatible with semaphore-based arbitration.

Srini discloses arbitration using arbiter cells implemented as a Mealy Machine using J-K flip-flops. Srini, col. 5, lines 63-66. In Srini, the arbitration scheme operates to select between requesters in a manner that ensures fairness by switching access between requesters in successive cycles. *See* Srini, col. 6, lines 5-33. Thus, Srini's arbitration includes forced alternation of access rights between different processors for successive cycles.

In contrast, as will be appreciated by one skilled in the art, semaphore-based arbitration permits a requester to lock a resource and use the resource until completion of an operation or series of operation[1]. Even using the limited definition of semaphores provided by the Examiner,

[1] *See, e.g.*, http://bama.ua.edu/cgi-bin/man-cgi?sem_wait+3RT and "Thread And Semaphore Examples," http://cse.stanford.edu/class/cs107/ handouts/32ConcurrencyExamples.pdf.

the use of prioritizing semaphores in Srini's system would defeat the fairness objectives of Srini. *See* Office Action of 11/9/05, page 3, first paragraph and Srini at col. 5, lines 21-35. Consequently, no motivation could have existed to combine and modify Srini and Holt to add a semaphore-based arbitration scheme that would necessarily conflict with Srini's fairness based scheme.

More particularly, Srini is directed to a crossbar switch that provides individual connections for processors and memory (*see* Fig. 1 and col. 3, line 59 – col. 4, line 10). Srini teaches a system that prevents more than one processor from accessing the same memory module. Srini at col. 2, lines 1-10. Srini teaches an arbiter for controlling access to a shared memory. *See, e.g.*, Srini at col. 5, lines 21-35. Srini explicitly teaches access control based on an arbiter that is constructed as a tree of one-of-two arbiters employing a synchronous scheme (Srini at col. 5, lines 36-66). Srini teaches the importance of such a scheme in providing an arbiter that "must be impartial in the sense that it gives equal priority to the processors ..." and Srini further teaches that "no single processor can dominate a memory module while others are waiting for the same module" (col. 5, lines 17-21). Thus, Srini implements a preemptive scheme that dictates alternating access rather than a semaphore locked until released scheme in which connections are permitted only when a resource is indicated as being available.

Although Srini teaches systems that have no need for semaphores, the Examiner nevertheless alleges that motivation could have existed to replace or somehow supplement Srini's arbiter with a hardware semaphore controller. Specifically, the Examiner offers that "Srini does not state how it knows which resource is in use" and "[p]reventing contention is the purpose of a semaphore that indicates which resource is in use." Office Action of 11/9/05, paragraph spanning pages 8 and 9. The Examiner is wrong. Srini's resources are connected at the direction of a state-based arbiter and arbitration requires knowledge of which requester is using each resource in successive states. *See* Srini, col. 6, lines 5-33. Simply put, an impartial Srini arbiter is aware of the status of all resources and the fairness of arbitration taught by Srini is explicitly predicated on such awareness. Therefore, the Examiner erred in believing that Srini lacked a means to prevent "multiple concurrent accesses of the same shared resource," and that semaphores would be of use in the fairness based state-machine arbiter taught by Srini.

Because Holt does not cure these basic deficiencies of Srini, the rejections should be reversed.

Goodwin And Holt Do Not Render Obvious A Resource Controller That Includes A Hardware Semaphore Unit For Controlling Access To Shared Resources

The combination of Goodwin and Holt does not render the claims obvious because Goodwin is directed to different subject matter in which a hardware semaphore unit for controlling access to shared resources is of no consequence. Goodwin teaches methods of maintaining cache coherency in systems where multiple processors access shared memory. Goodwin mentions in passing that an arbiter chip 14 decides which of plural CPUs could have access to data that happened to be in the same memory module. Goodwin, col. 4, lines 49-65. However, operation of the arbiter chip 14 is not the subject of the Goodwin disclosure and is not described or discussed further. Instead Goodwin is directed to methods for ensuring cache coherency in a system arbitrated by an arbiter chip 14. *See, e.g.*, Goodwin Abstract.

Goodwin is directed to methods for improving data coherency in cache systems. Goodwin, paragraph spanning cols. 3 and 4. Goodwin teaches the mapping of data bits to guarantee that fill and victim data are from the same memory module. *Id.* Nothing in Goodwin requires the use of a semaphore or the control of access to shared resources by multiple processors. Goodwin does not address arbitration schemes and is silent regarding controlling access to resources. Further, Goodwin is silent regarding use of semaphores to control access to resources. Accordingly, Goodwin cannot be said to teach or suggest a resource controller that includes a hardware semaphore unit for controlling access to the shared resources as required in the claims because Goodwin is unconcerned with access control and directed only to managing the consequences of arbitration.

Thus, Goodwin offers no grounds that can be reasonably be used to reject the claims of the present application and a skilled artisan would not have been motivated to combine Goodwin with other cited art in any manner that renders the claimed inventions obvious.

Therefore, the rejections should be reversed.

Hiller And Holt Do Not Render Obvious A Resource Controller That Includes A Hardware Semaphore Unit For Controlling Access To Shared Resources

The combination of Hiller and Holt does not render the claims obvious because Hiller is directed to different subject matter in which a hardware semaphore unit would be of no use. Hiller is silent regarding use of semaphores to control access to resources. Further, Hiller teaches ***elimination*** of the need for arbitration. Specifically, Hiller describes a system in which crossbar switch configuration is predetermined at the time that microcode algorithms are developed, thereby eliminating the need for arbitration of access to parallel memories during run time. Hiller, paragraph spanning cols. 6 and 7. Hiller obviates the use of semaphore controlled access to shared resources because contention is eliminated. Thus nothing in Hiller requires the use of a semaphore or the control of access to shared resources by multiple processors. Accordingly, Hiller provides no grounds that can be reasonably be used to reject the claims of the present application. Furthermore, a skilled artisan would not have been motivated to combine Hiller with other cited art in any manner that renders the claimed inventions obvious and the claimed advantages of Hiller redundant.

Therefore, the rejections should be reversed.

No Combination Of The Cited Art Renders Obvious A Hardware Semaphore Unit

The Examiner suggests that hardware and software is logically equivalent and relies on this suggestion in rejecting the claims. However, the Examiner cites no art that teaches the implementation of a hardware semaphore unit. Nor does the Examiner articulate a reason that a skilled artisan would have been motivated to alter the cited art to implement a semaphore unit in hardware. The Examiner's argument is flawed and insufficient to support a rejection of the claims.

The Examiner generalizes that hardware and software are logically equivalent but provides no citation in support of the generalization. Office Action of November 9, 2005 at page 4, first paragraph. Next, the Examiner attempts to show motivation to alter references by speculating that well known purposes for using hardware include "hardware is faster than software." *Id.* (Notably, the references teach neither hardware nor software semaphore units) However, no citation is provided in support of this latter speculation and Appellants respectfully submit that at least an equal number of disincentives would have been apparent.

For example, Appellants note that, in systems of interest, multiple processors may have been available that would have been capable of executing portions of a software semaphore unit. Additionally, the Examiner has not given due consideration to relative clocking speeds of processor and speculative hardware semaphore unit (e.g. internal clock v. peripheral bus speed) that could have easily allowed a processor to implement a faster software semaphore scheme. In these examples, hardware may not have offered a speed advantage over software and the Examiner's argument fails. Consequently, it is unreasonable to assert existence of motivation to convert a semaphore unit to hardware when the references teach no kind of semaphore unit and, with respect to the cited art, no specific clear advantage to be accrued from conversion has been identified by the Examiner.

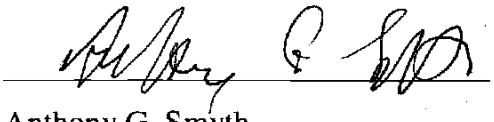
Appellants respectfully submit that the Examiner erred in rejecting the claims based on unsupported assertions offering questionable advantages. Therefore, the rejections should be reversed.

CONCLUSION

For the foregoing reasons, Appellants respectfully request that all the pending claims be deemed allowable by this honorable Board.

Date: July 10, 2006

PILLSBURY WINTHROP SHAW PITTMAN LLP
2475 Hanover Street
Palo Alto, CA 94304-1114
Telephone: (650) 233.4802
Facsimile: (650) 233-4545
Customer Number: 27498



Anthony G. Smyth
Registration No. 55,636

CLAIMS APPENDIX

1. (Previously presented) A computer system having a multipath cross bar bus, comprising:

one or more processors;

one or more resources capable of being shared by the one or more processors; and

a resource controller and bus that is connected to each resource and to each processor,

wherein the resource controller is capable of permitting each processor to simultaneously access a different resource from the one or more resources, and

wherein the resource controller includes a hardware semaphore unit for controlling access to the shared resources.

2. (Previously presented) The system of Claim 1, wherein the one or more resources further comprise one or more memory resources and wherein the resource controller further comprises a memory controller that is capable of permitting a first processor to access a first memory resource and a second processor to access a second memory resource at the same time.

3. (Original) The system of Claim 2, wherein the memory controller further comprises one or more switches that are capable of selecting a particular memory resource to connect to a particular processor and a resource arbitration controller that controls the one or more switches in order to dynamically connect each processor independently to each memory resource.

4. (Original) The system of Claim 3, wherein the one or more switches comprise one or more multiplexers.

5. (Original) The system of Claim 4, wherein the resources further comprise one or more peripheral resources and wherein the resource controller further comprises a peripheral controller that is capable of permitting a first processor to access a first peripheral resource and a second processor to access a second peripheral resource at the same time.

6. (Original) The system of Claim 5, wherein the peripheral controller further comprises one or more switches that are capable of selecting a particular peripheral resource to connect to a particular processor and a resource arbitration controller that controls the one or more switches in order to dynamically connect each processor independently to each peripheral resource.

7. (Original) The system of Claim 6, wherein the one or more switches comprise one or more multiplexers.

8. (Original) The system of Claim 1, wherein the resources further comprise one or more peripheral resources and wherein the resource controller further comprises a peripheral controller that is capable of permitting a first processor to access a first peripheral resource and a second processor to access a second peripheral resource at the same time.

9. (Original) The system of Claim 8, wherein the peripheral controller further comprises one or more switches that are capable of selecting a particular peripheral resource to connect to a particular processor and a resource arbitration controller that controls the one or more switches in order to dynamically connect each processor independently to each peripheral resource.

10. (Original) The system of Claim 9, wherein the one or more switches comprise one or more multiplexers.

11. (Previously presented) An apparatus for controlling the access to one or more computing resources by one or more processors, the apparatus comprising a resource controller and bus that is connected to each resource and to each processor wherein the resource controller is capable of permitting each processor to simultaneously access a different resource from the one or more resources, and wherein the resource controller includes a hardware semaphore unit for controlling access to the one or more resources.

12. (Original) The apparatus of Claim 11, wherein the resources further comprise one or more memory resources and wherein the resource controller further comprises a memory controller that is capable of permitting a first processor to access a first memory resource and a second processor to access a second memory resource at the same time.

13. (Original) The apparatus of Claim 12, wherein the memory controller further comprises one or more switches that are capable of selecting a particular memory resource to connect to a particular processor and a resource arbitration controller that controls the one or more switches in order to dynamically connect each processor independently to each memory resource.

14. (Original) The apparatus of Claim 13, wherein the one or more switches comprise one or more multiplexers.

15. (Original) The apparatus of Claim 14, wherein the resources further comprise one or more peripheral resources and wherein the resource controller further comprises a peripheral controller that is capable of permitting a first processor to access a first peripheral resource and a second processor to access a second peripheral resource at the same time.

16. (Original) The apparatus of Claim 15, wherein the peripheral controller further comprises one or more switches that are capable of selecting a particular peripheral resource to connect to a particular processor and a resource arbitration controller that controls the one or more switches in order to dynamically connect each processor independently to each peripheral resource.

17. (Original) The apparatus of Claim 16, wherein the one or more switches comprise one or more multiplexers.

18. (Original) The apparatus of Claim 11, wherein the resources further comprise one or more peripheral resources and wherein the resource controller further comprises a peripheral controller that is capable of permitting a first processor to access a first peripheral resource and a second processor to access a second peripheral resource at the same time.

19. (Original) The apparatus of Claim 18, wherein the peripheral controller further comprises one or more switches that are capable of selecting a particular peripheral resource to connect to a particular processor and a resource arbitration controller that controls the one or more switches in order to dynamically connect each processor independently to each peripheral resource.

20. (Original) The apparatus of Claim 19, wherein the one or more switches comprise one or more multiplexers.

21. (Previously presented) An apparatus for controlling the access to one or more memory resources by one or more processors, the controller comprising a memory resource controller and bus that is connected to each memory resource and to each processor, wherein the resource controller includes a hardware semaphore unit for controlling access to the shared resources, and wherein the memory resource controller is capable of permitting each processor to simultaneously access a different resource from the one or more memory resources.

22. (Previously presented) The apparatus of Claim 21, wherein the memory controller further comprises one or more switches that are capable of selecting a particular memory resource to connect to a particular processor and a resource arbitration controller that controls the one or more switches in order to dynamically connect each processor independently to each memory resource.

23. (Previously presented) The apparatus of Claim 22, wherein the one or more switches comprise one or more multiplexers.

24. (Previously presented) An apparatus for controlling access by one or more processors to one or more peripheral resources, the apparatus comprising a peripheral resource controller and bus that is connected to each peripheral resource and to each processor, wherein the resource controller is capable of permitting each processor to simultaneously access a different peripheral resource from the one or more peripheral resources, wherein the peripheral resource controller includes a hardware semaphore unit for controlling access to the one or more peripheral resources, the hardware semaphore unit being configured to receive requests from the one or more processors and prioritize access based on the requests.

25. (Original) The controller of Claim 24, wherein the peripheral controller further comprises one or more switches that are capable of selecting a particular peripheral resource to connect to a particular processor and a resource arbitration controller that controls the one or more switches in order to dynamically connect each processor independently to each peripheral resource.

26. (Original) The controller of Claim 25, wherein the one or more switches comprise one or more multiplexers.

27. (Previously presented) A computer system, comprising:

a first processor capable of executing a set of instructions;

a second processor capable of executing a set of instructions;

a multipath memory controller having a first bus that is capable of connecting the first processor to a set of memory resources and a second bus that is capable of connecting the second processor to the same set of memory resources wherein the first and second processors are capable of simultaneously accessing different memory resources, wherein the multipath memory controller includes a first semaphore unit for prioritizing access to the set of memory resources; and

a multipath peripheral controller having a first bus that is capable of connecting the first processor to a set of peripheral resources and a second bus that is capable of connecting the second processor to the same set of peripheral resources wherein the first and second processors are capable of simultaneously accessing different peripheral resources, wherein the multipath peripheral controller includes a second semaphore unit for prioritizing access to the set of peripheral resources, wherein

at least one of the semaphore units comprises a hardware semaphore unit.

APPENDIX: EVIDENCE

None.

APPENDIX: RELATED PROCEEDINGS

None.